B104 포팅 매뉴얼 (빌드, 배포)

팀명 : 선보고 후조치

프로젝트명 : Get Idea

백엔드(BE)

Java: OpenJDK 17.0.1, Spring Boot: 3.2.2 (내장 웹서버 사용), Gradle: 8.5, JPA: 3.2

Lombok, Spring Security: 6.2, OAuth2 JWT: 6.2, Node.js: 20.0.1, Express.js: 4.18.2, Swagger: 2.2

프론트엔드(FE)

React: 18.2, Yarn: 1.22.19, Node.js: 20.0.1, Tailwind CSS: 3.0.2, TypeScript: 3.9.10, JavaScript: ES6

SASS

데이터베이스(DB)

MySQL: 8.0, Redis: 3.2, MongoDB: 4.2

DevOps 및 추가 도구

Jenkins: 2.426.3, Nexus Repository, Docker: 19.03.12

웹서버 및 기타 기술

Nginx 1.18.0, WebSocket ,WebRTC

개발 환경

IntelliJ IDEA: 2023.3.4 (Ultimate Edition), Visual Studio Code: 1.86

환경 변수

.env

REACT\_APP\_LOGIN\_REDIRECT\_URI=https://i10b104.p.ssafy.io&mode=login

REACT\_APP\_UNLINK\_REDIRECT\_URI=https://i10b104.p.ssafy.io&mode=unlink

REACT\_APP\_GOOGLE\_LOGIN\_URI=https://i10b104.p.ssafy.io/oauth2/authorization/google

REACT\_APP\_KAKAO\_LOGIN\_URI=https://i10b104.p.ssafy.io/oauth2/authorization/kakao

REACT\_APP\_NAVER\_LOGIN\_URI=https://i10b104.p.ssafy.io/oauth2/authorization/naver

REACT\_APP\_API\_URI=https://i10b104.p.ssafy.io

REACT\_APP\_WEBSOCKET\_URL=wss://i10b104.p.ssafy.io:8000

REACT\_APP\_PEERJS\_URL=https://i10b104.p.ssafy.io/peer

Application.yml

spring:  
 jackson:  
 serialization:  
 write-dates-as-timestamps: false  
  
 data:  
 redis:  
 host: i10b104.p.ssafy.io  
 port: 6380  
  
  
 mongodb:  
 uri: mongodb://gimongos10b104:bs970610yg980512@i10b104.p.ssafy.io:27017/gimongo  
  
 datasource:  
 driver-class-name: com.mysql.cj.jdbc.Driver  
 jdbc-url: jdbc:mysql://i10b104.p.ssafy.io:3306/gimysql?useSSL=false&serverTimezone=UTC&allowPublicKeyRetrieval=true  
 username: root  
 password: jk950905hs970114  
  
 thymeleaf:  
 cache: false  
  
 jpa:  
 hibernate:  
 ddl-auto: update  
 show-sql: true  
 open-in-view: false  
 *#DB Table의 Column명과 변수명을 그대로 하고 싶을 때 사용하는 설정* naming:  
 physical-strategy: org.hibernate.boot.model.naming.PhysicalNamingStrategyStandardImpl  
  
  
 *#OAuth 2.0 소셜 로그인 관련 설정* security:  
 oauth2:  
 client:  
 registration:  
 google:  
 clientId: # google client id   
 clientSecret: # google client secret  
 scope:  
 - email  
 - profile  
 kakao:  
 clientId: # kakao client-id

clientSecret: # kakao client-secet  
 client-authentication-method: client\_secret\_post  
 authorization-grant-type: authorization\_code  
 redirectUri: "{baseUrl}/{action}/oauth2/code/{registrationId}"  
 scope: *# https://developers.kakao.com/docs/latest/ko/kakaologin/common#user-info* - profile\_nickname  
 - profile\_image  
 - account\_email  
 client-name: Kakao  
  
 naver:  
 client-id: # naver client-id  
 client-secret: # naver client-secret  
 client-authentication-method: client\_secret\_post  
 authorization-grant-type: authorization\_code  
 redirect-uri: "{baseUrl}/{action}/oauth2/code/{registrationId}"  
 *# scope:  
 # - name  
 # - email  
 # - profile\_image* client-name: Naver  
 provider:  
 kakao:  
 authorization-uri: https://kauth.kakao.com/oauth/authorize  
 token-uri: https://kauth.kakao.com/oauth/token  
 user-info-uri: https://kapi.kakao.com/v2/user/me  
 user-info-authentication-method: header  
 user-name-attribute: id *# Kakao 응답 값 id, connected\_at, properties, kakao\_account 중 id 지정* naver:  
 authorization-uri: https://nid.naver.com/oauth2.0/authorize  
 token-uri: https://nid.naver.com/oauth2.0/token  
 user-info-uri: https://openapi.naver.com/v1/nid/me  
 user-info-authentication-method: header  
 user-name-attribute: response *# Naver 응답 값 resultCode, message, response 중 response 지정* cloud:  
 aws:  
 s3:  
 bucket: getidea-s3  
 *# folder:  
 # [VARIABLE]: [VALUE]* credentials:  
 access-key: # S3 access-key

secret-key: # S3 secret-key  
 region:  
 static: ap-northeast-2  
 auto: false  
 stack:  
 auto: false  
 servlet:  
 multipart:  
 enabled: true  
 max-file-size: 10MB  
 max-request-size: 10MB  
  
jwt: *# base64로 인코딩된 512비트(64바이트) 이상의 키* access:  
 expiration: 3600000 *# 1시간* header: Authorization  
 refresh:  
 expiration: 86400000 *# 1일* header: Authorization-refresh  
  
server:  
 port: 8084  
  
logging:  
 file:  
 name: /var/log/myapp.log

Nginx 설정 파일

server {

listen 80 default\_server;

listen [::]:80 default\_server;

root /var/www/html;

index index.html index.htm index.nginx-debian.html;

server\_name \_;

location / {

try\_files $uri $uri/ =404;

}

server {

root /home/jenkins/workspace/Getidea/GetiDEA/build;

index index.html index.htm;

server\_name i10b104.p.ssafy.io; # managed by Certbot

location / {

try\_files $uri $uri/ /index.html;

}

location /api/ {

proxy\_pass http://localhost:8084/api/;

proxy\_set\_header Host $host;

proxy\_set\_header X-Real-IP $remote\_addr;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

proxy\_set\_header X-Forwarded-Proto $scheme;

proxy\_redirect off;

proxy\_buffering off;

proxy\_http\_version 1.1;

proxy\_set\_header Upgrade $http\_upgrade;

proxy\_set\_header Connection "Upgrade";

}

location /openvidu/ {

proxy\_pass http://localhost:4443/;

proxy\_set\_header Host $host;

proxy\_set\_header X-Real-IP $remote\_addr;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

proxy\_set\_header X-Forwarded-Proto $scheme; proxy\_redirect off;

proxy\_buffering off;

proxy\_http\_version 1.1;

proxy\_set\_header Upgrade $http\_upgrade;

proxy\_set\_header Connection "Upgrade";

}

location /login/ {

proxy\_pass http://localhost:8084/login/;

proxy\_set\_header Host $host;

proxy\_set\_header X-Real-IP $remote\_addr;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

proxy\_set\_header X-Forwarded-Proto $scheme;

proxy\_buffering off;

proxy\_http\_version 1.1;

proxy\_set\_header Upgrade $http\_upgrade;

proxy\_set\_header Connection "Upgrade";

}

location /peer/ {

proxy\_pass http://localhost:5000/;

proxy\_set\_header Host $host;

proxy\_set\_header X-Real-IP $remote\_addr;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

proxy\_set\_header X-Forwarded-Proto $scheme;

proxy\_redirect off;

proxy\_buffering off;

proxy\_http\_version 1.1;

proxy\_set\_header Upgrade $http\_upgrade;

proxy\_set\_header Connection "Upgrade";

}

location /oauth2/ {

proxy\_pass http://localhost:8084/oauth2/;

proxy\_set\_header Host $host;

proxy\_set\_header X-Real-IP $remote\_addr;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

proxy\_set\_header X-Forwarded-Proto $scheme;

proxy\_redirect off;

proxy\_buffering off;

proxy\_http\_version 1.1;

proxy\_set\_header Upgrade $http\_upgrade;

proxy\_set\_header Connection "Upgrade";

}

location /jenkins {

proxy\_pass http://localhost:8082/jenkins;

proxy\_set\_header Host $host;

proxy\_set\_header X-Real-IP $remote\_addr;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

proxy\_set\_header X-Forwarded-Proto $scheme;

proxy\_redirect off;

proxy\_buffering off;

proxy\_http\_version 1.1;

proxy\_set\_header Upgrade $http\_upgrade;

proxy\_set\_header Connection "Upgrade";

}

listen [::]:443 ssl ipv6only=on; # managed by Certbot

listen 443 ssl; # managed by Certbot

ssl\_certificate /etc/letsencrypt/live/i10b104.p.ssafy.io/fullchain.pem; # managed by Certbot

ssl\_certificate\_key /etc/letsencrypt/live/i10b104.p.ssafy.io/privkey.pem; # managed by Certbot

include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot

ssl\_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot

}

server {

if ($host = i10b104.p.ssafy.io) {

return 301 https://$host$request\_uri;

} # managed by Certbot

listen 80 ;

listen [::]:80 ;

server\_name i10b104.p.ssafy.io;

location / {

root /var/www/html;

index index.html;

try\_files $uri $uri/ =404;

}

}